



Image Classification: Clothing Dataset

Li Cao, Meghna Diwan,
Viviana Hernandez, Vamika Venkatesan

Agenda

Problem Statement (Meghna)

Assumptions/Hypotheses (Meghna) ?

Data Properties & Transformations *EDA & visualization (this inspires us some assumptions of models) (Meghna)

Proposed Approaches (Models) * ML, NN, CNN, Transfer Learning (Li)

Proposed Solution (Model Selection with visualizations) * compare accuracy + F1, with respect of visualization , a table form (Viv & Vams)

Results (Accuracy) (probably involves metrics such as overfitting / underfitting) * reduce epoches (Viv & Vams)

Future Work * knowledge from transfer learning

Team Introduction



Li Cao



Meghna Diwan



Viviana Hernandez



Vamika Venkatesan

A photograph of a woman from the waist up. She is wearing a light blue denim blouse with ruffled sleeves and a belt, paired with white pants and a belt. She is standing in front of a window with white blinds. A small portion of a sign that says "FC" is visible in the background.

Agenda

Background

Data Properties

Models Implemented

Final Results

Lessons Learned

Future Work



Image Classification & E-commerce

\$481.2 b

Worldwide Revenue from
Ecommerce Fashion Industry in 2018

Need for Image Classification



Helps consumers with visual search making print ads more shoppable and providing easy access to lookalikes



Optimizes search by automating tagging and categorizing leading to increase in sales



Better ad targeting for consumer by using photos shared on social media



Data Properties

No. of Images: 70,000 labelled

Size: 28x28

Classes: 10 (Balanced)

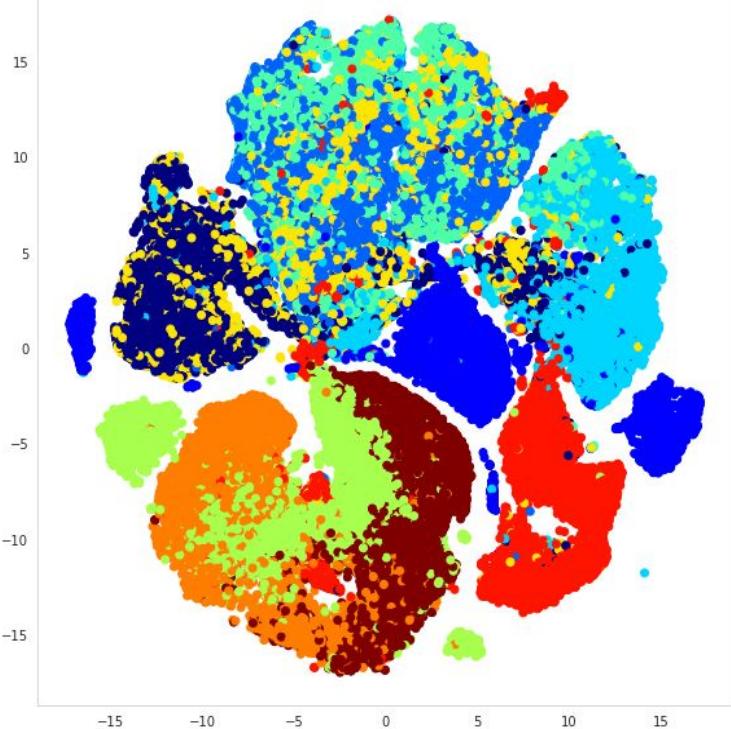
Pixel Values: $[0, 255] \rightarrow [0,1]$



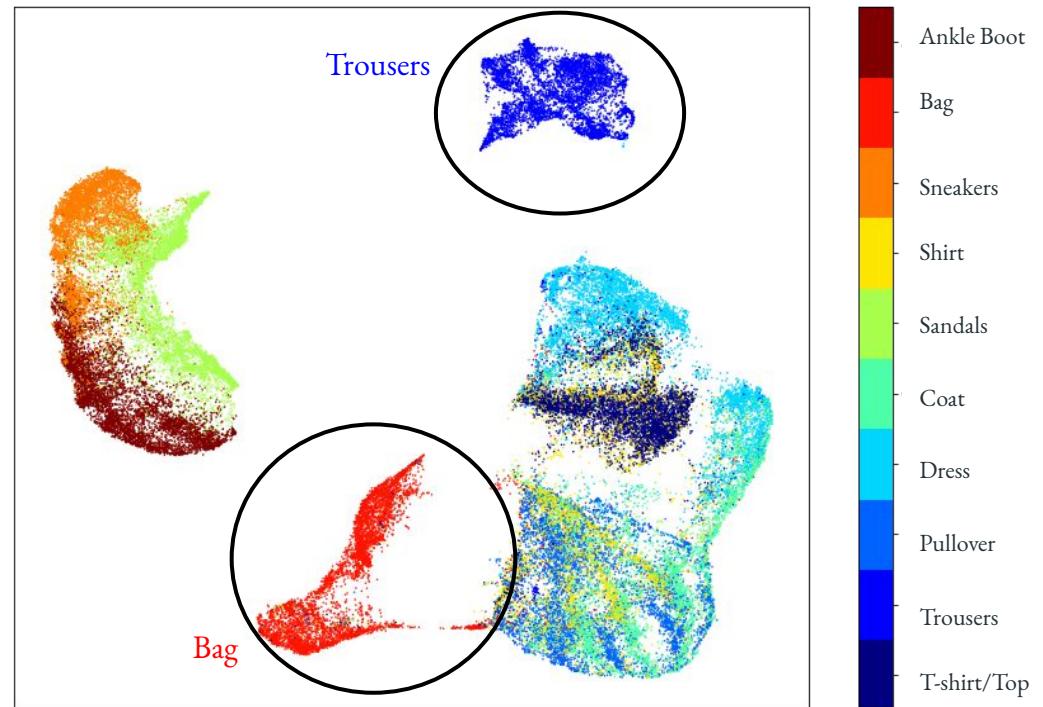
					→ T-shirt/Top
					→ Trouzers
					→ Pullover
					→ Dress
					→ Coat
					→ Sandals
					→ Shirt
					→ Sneakers
					→ Bag
					→ Ankle Boot

Visualize in 2-D

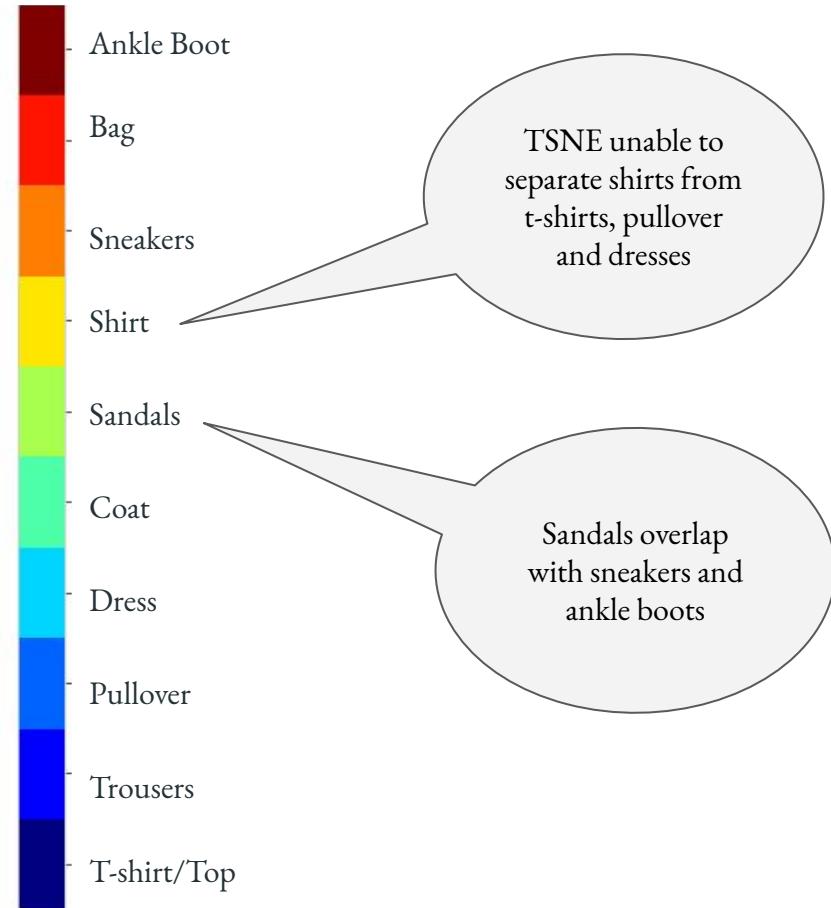
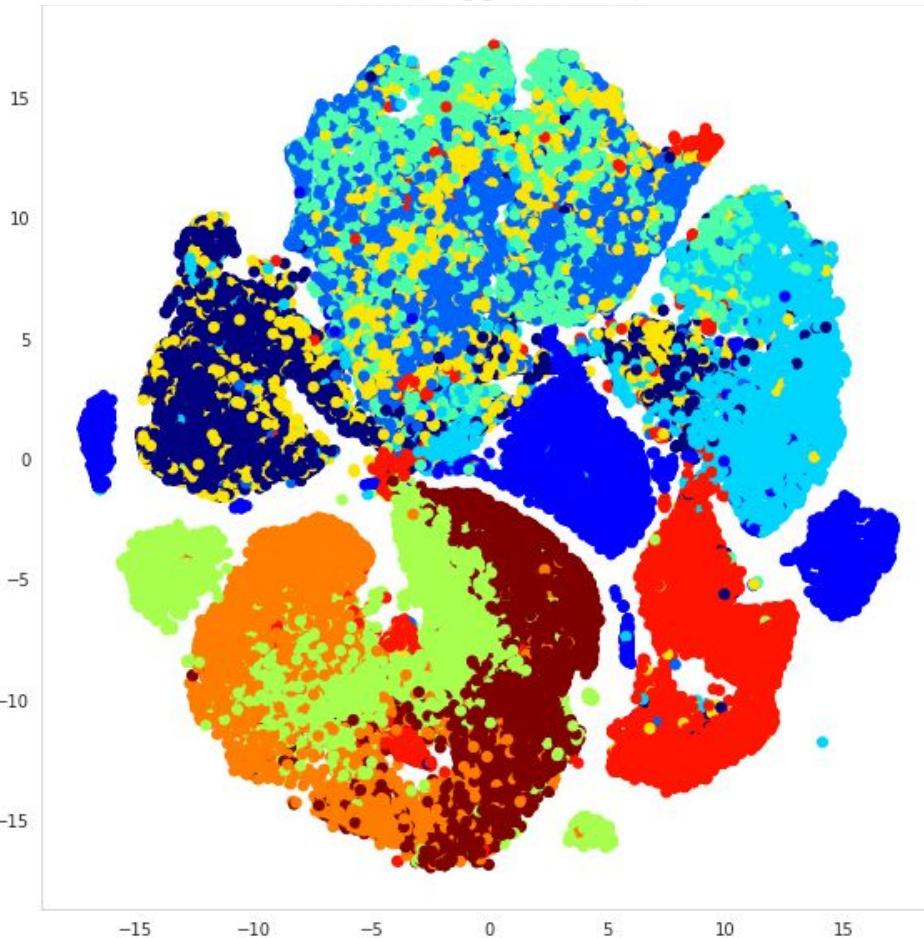
Clustering Apparel Using **T-SNE**



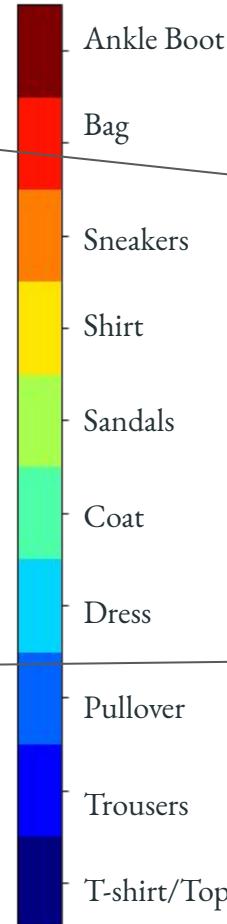
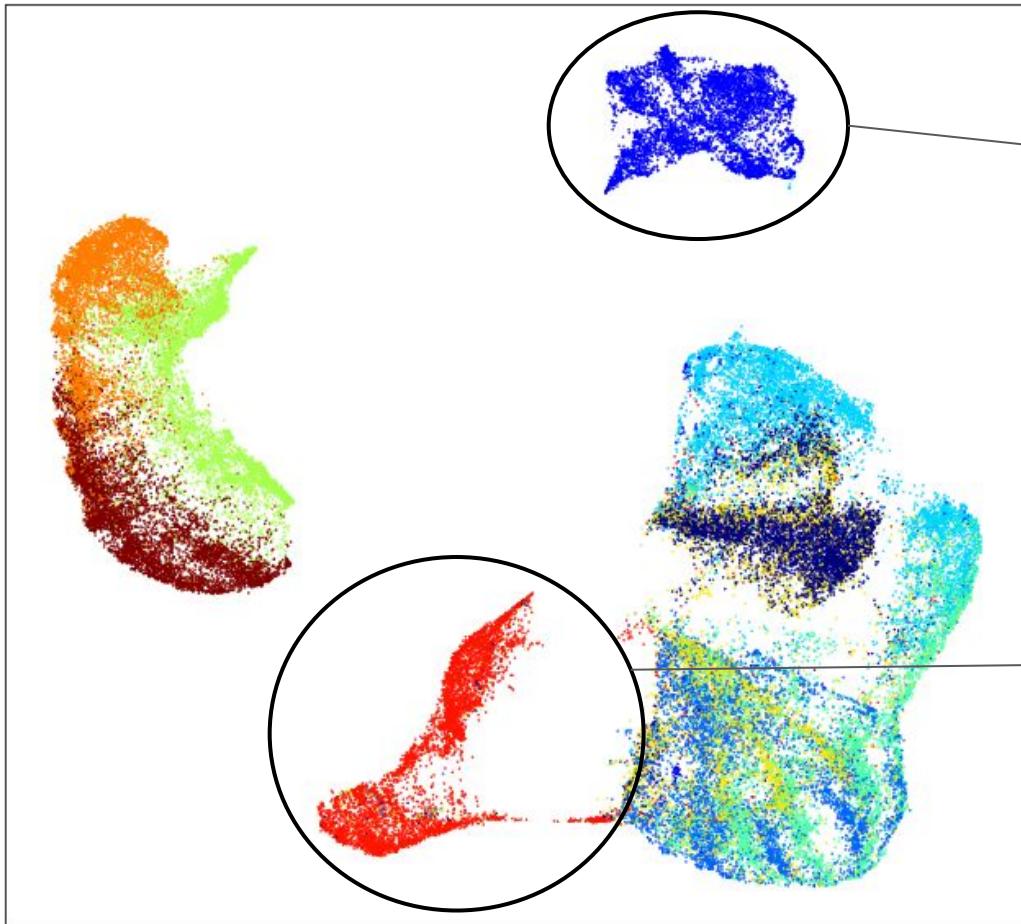
Clustering Embedding Using **UMap**



Clustering Apparel Using T-SNE



Clustering Embedding Using UMap



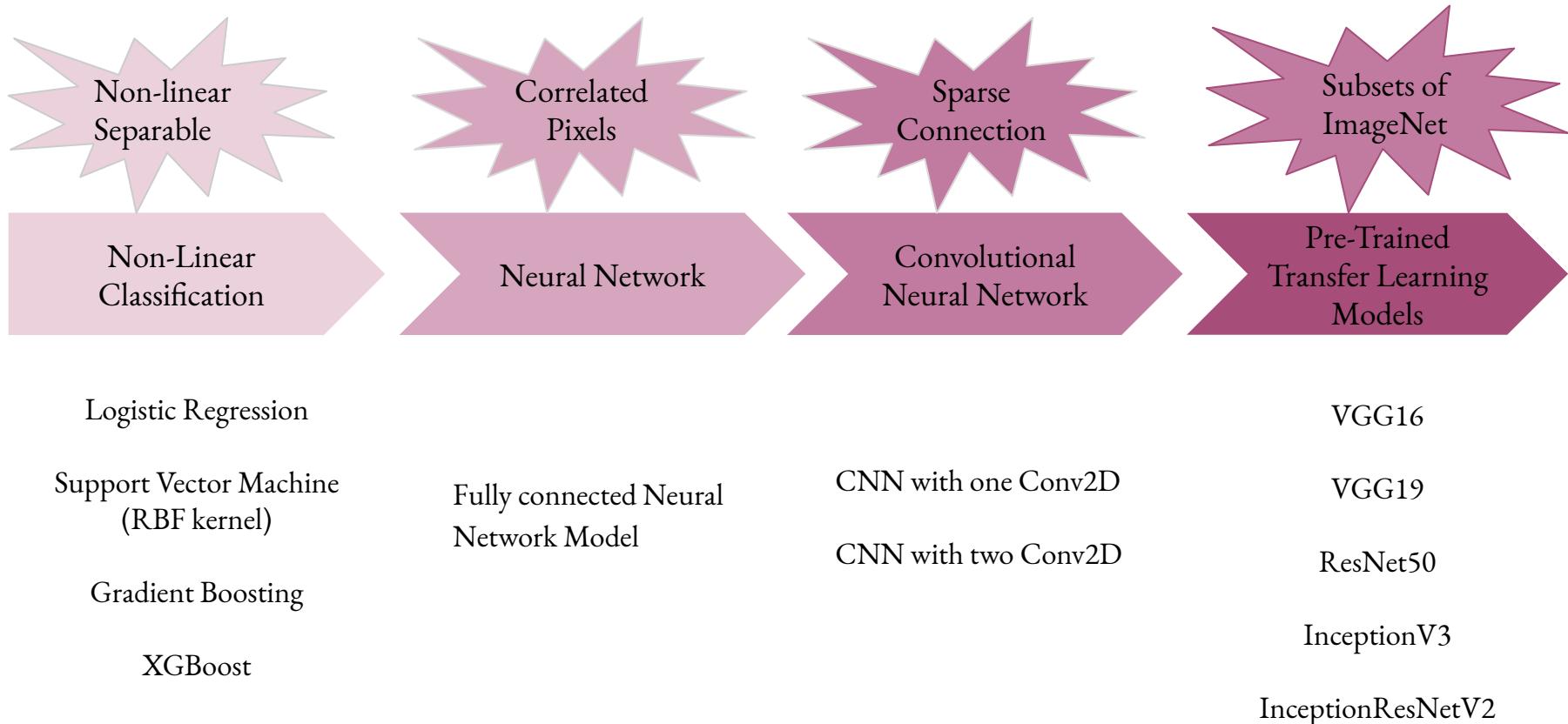
Trousers are easily separable because of their distinct shape

Bag are also distinct from the rest of the classes

Models Implemented



Model Techniques



Models Limitations & Solutions

Overfitting

Traditional Machine Learning techniques & Neural Network & CNN tend to learn too detailed about train set with 60,000 rows, generalization error is large



Hyperparameter tuning such as dropout, learning rate, early stopping and image rotations, etc.

Underfitting

Pre-trained models are build on rich data, our train set is too simple to be aborted well, so transfer learning tends to underfit in our case



Increase data richness and complexity by expanding our image pixel size

Models Limitations & Solutions

Limitation:

Slack performance for some class due to similarity, not imbalance issue



Model Performance

Deep Learning

91.9%

Average Accuracy

No Overfitting

78%

Highest recall for Shirts

Two Conv2D

93%

One Conv2D

90.85%

88.5%

Average Accuracy

Overfitting

80%

Highest recall for Shirts

Machine Learning

86.8%

Average Accuracy

Overfitting

72%

Highest recall for Shirts

Neural Networks

88.5%

Gradient Boosting

90%

SVM

87.24%

Logistic Regression

85.25%

XGBoost

84.86%

Transfer Learning

%

Average Accuracy

Underfitting

InceptionResNetV2

%

InceptionV3

%

ResNet50

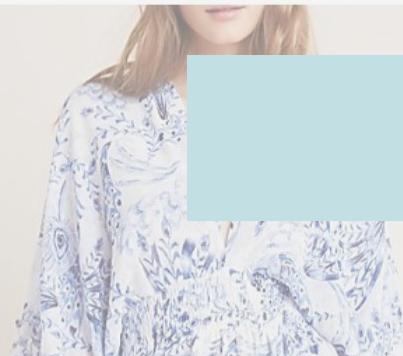
%

VGG16

10 %

VGG19

10 %



Evaluation of Machine Learning Models

Support Vector Machine

87.2%

Average Accuracy Score

Precision Score: 87.1%
Recall Score: 87.2%
F1 Score: 87.1%



- Overfitting: Yes
- Best: Trouser (98.0%)
- Worst: Shirt (66.0%)

eXtreme Gradient Boosting

84.9%

Average Accuracy Score

Precision Score: 84.8%
Recall Score: 84.9%
F1 Score: 84.7%



- Overfitting: Yes
- Best: Trouser (97.0%)
- Worst: Shirt (61.0%)

Logistic Regression

85.3%

Average Accuracy Score

Precision Score: 85.0%
Recall Score: 85.3%
F1 Score: 85.1%



- Overfitting: Yes
- Best: Trouser (97.0%)
- Worst: Shirt (62.0%)

Gradient Boosting Classifier

89.5%

Average Accuracy Score

Precision Score: 89.5%
Recall Score: 89.5%
F1 Score: 89.5%



- Overfitting: Yes
- Best: Trouser (98.0%)
- Worst: Shirt (72.0%)

Evaluation of Deep Learning Models

Simple Neural Network

89.0%

Average Accuracy Score

Precision Score: 89.0%

Recall Score: 89.0%

F1 Score: 89.0%



- Overfitting: Yes
- Best: Trouser (98.0%)
- Worst: Shirt (71.0%)

One Conv2D Layer

91.2%

Average Accuracy Score

Precision Score: 91.5%

Recall Score: 91.2%

F1 Score: 91.2%



- Overfitting: Yes
- Best: Trouser, Sandal, Bag (98.0%)
- Worst: Shirt (76.0%)

Two Conv2D Layers

92.6%

Average Accuracy Score

Precision Score: 92.5%

Recall Score: 92.6%

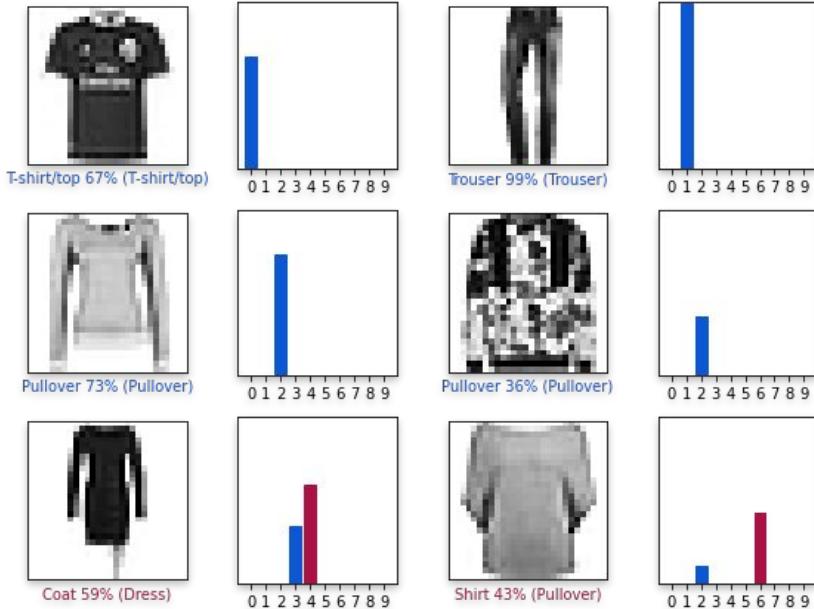
F1 Score: 92.5%



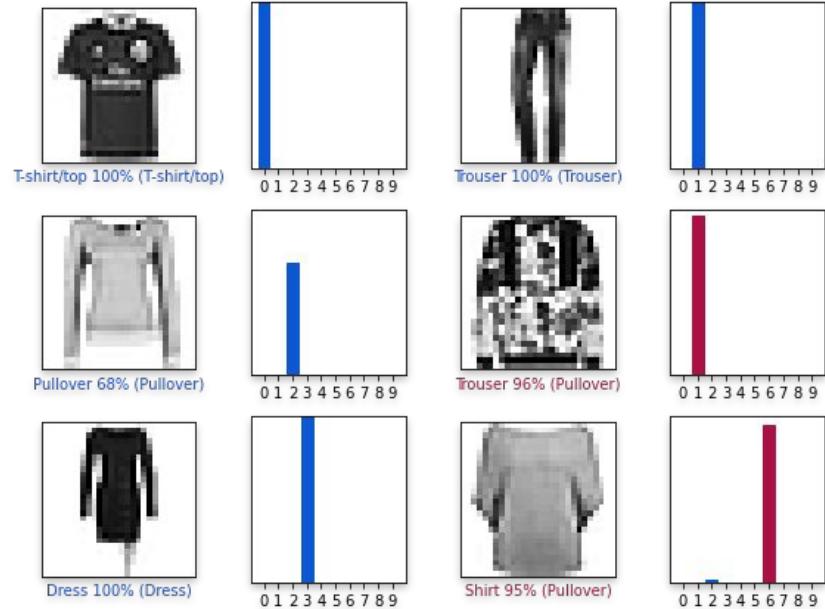
- Overfitting: Yes
- Best: Trouser (99.0%)
- Worst: Shirt (79.0%)

ML Model Prediction vs DL Model Prediction

Logistic Regression



Two Conv2D Layers

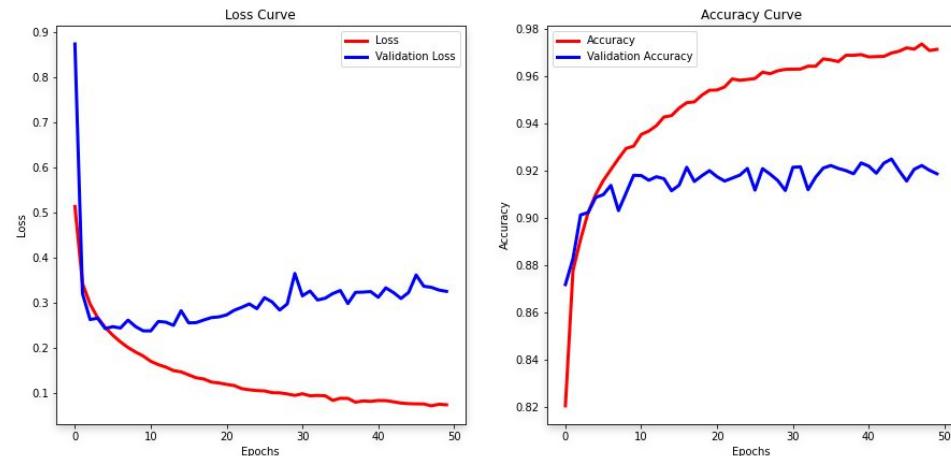


Performance of Final Model - Two Conv2D Layers

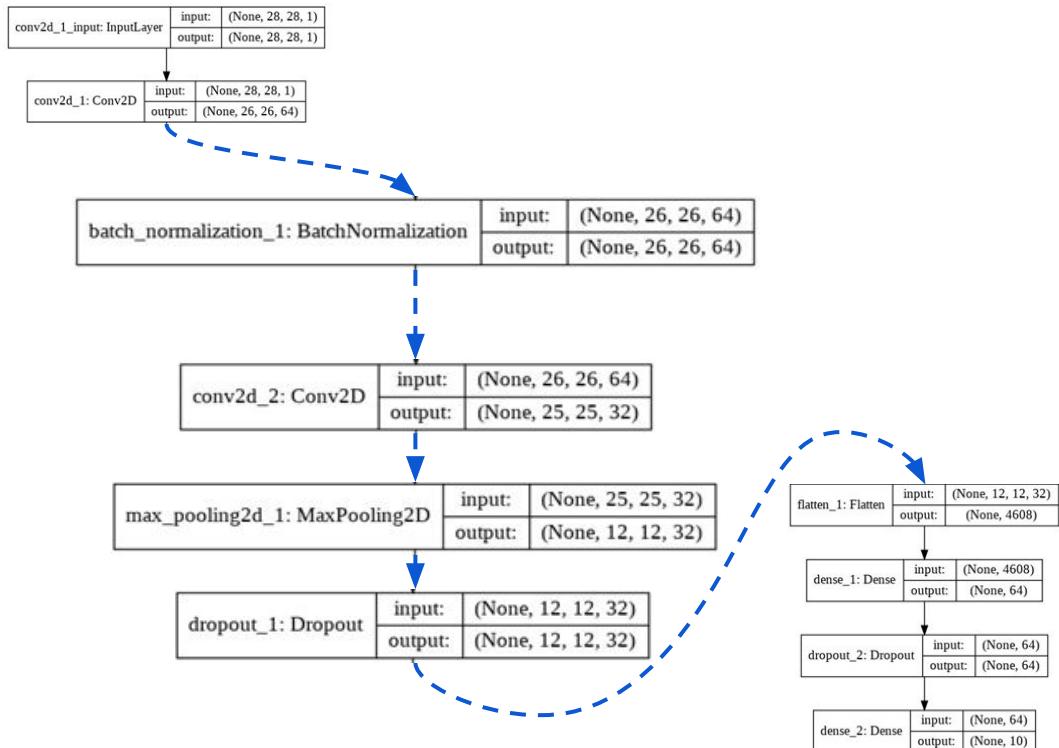
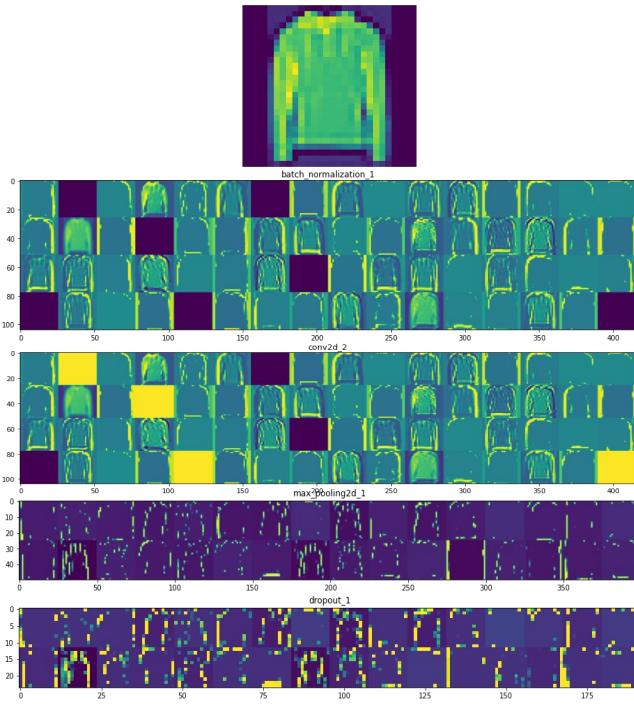
Classification Report

CNN 2 (Test)				
	precision	recall	f1-score	support
T-shirt/top	0.84	0.88	0.86	1000
Trouser	0.99	0.99	0.99	1000
Pullover	0.92	0.85	0.89	1000
Dress	0.93	0.93	0.93	1000
Coat	0.88	0.92	0.90	1000
Sandal	0.99	0.97	0.98	1000
Shirt	0.80	0.79	0.79	1000
Sneaker	0.97	0.95	0.96	1000
Bag	0.98	0.99	0.98	1000
Ankle boot	0.95	0.98	0.97	1000
accuracy			0.93	10000
macro avg	0.93	0.93	0.93	10000
weighted avg	0.93	0.93	0.93	10000

Loss / Accuracy Plots

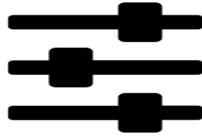


Performance of Final Model - Two Conv2D Layers



Lessons Learned





Tuning Parameters using trial and error required large computational power especially in the case of Transfer Learning Models. Knowledge of the architecture specifics would have helped.

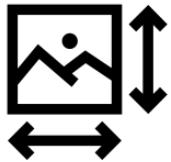
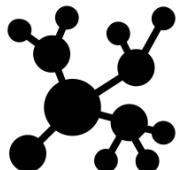


Image Size was a significant factor in the richness of features extracted from Transfer Learning Models. With larger images, the networks were able to extract better features and hence predicted better.



Increasing depth of the network did not increase performance of the models. Often, adding model complexity only increased the probability of overfitting.

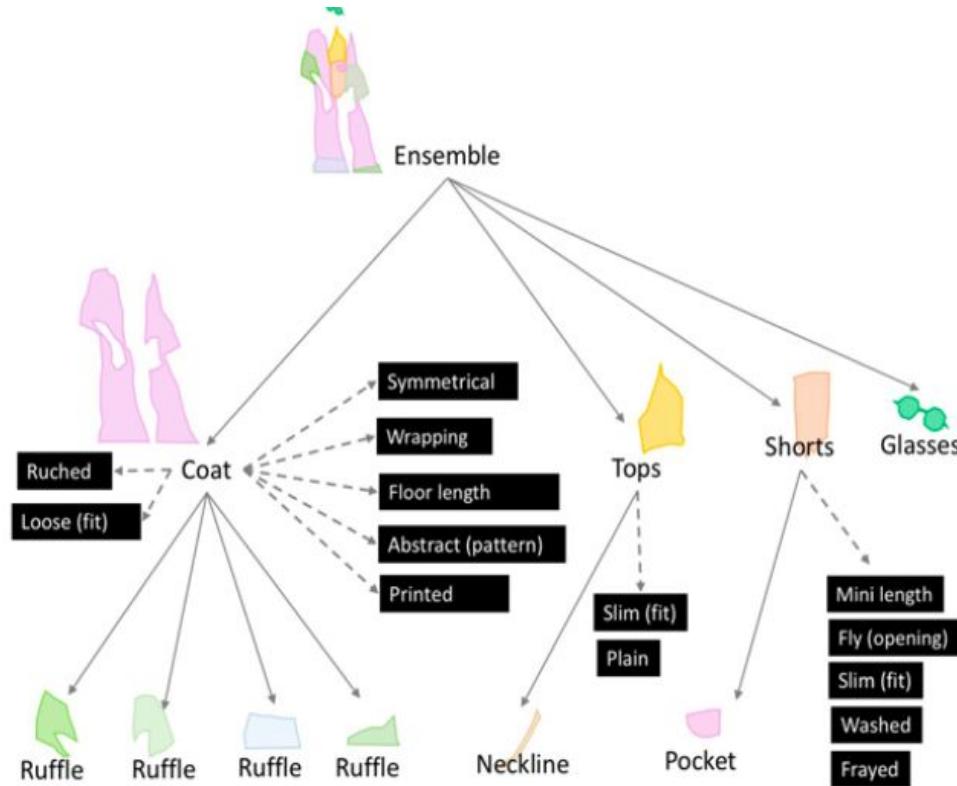


Thank You



Future Work

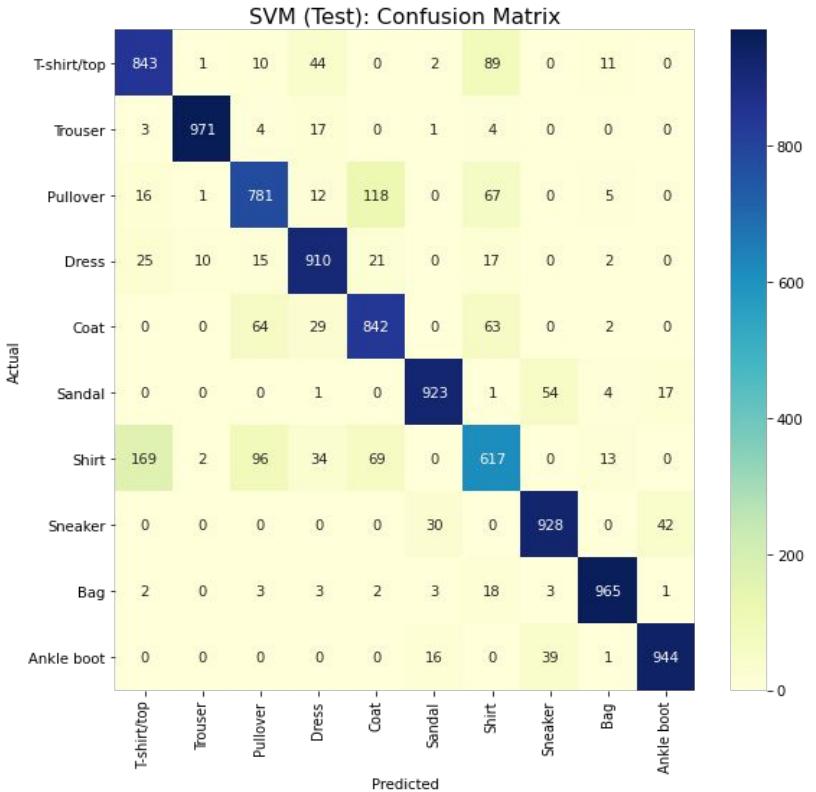
Expand to Multiclass Classification and Image Detection



Appendix



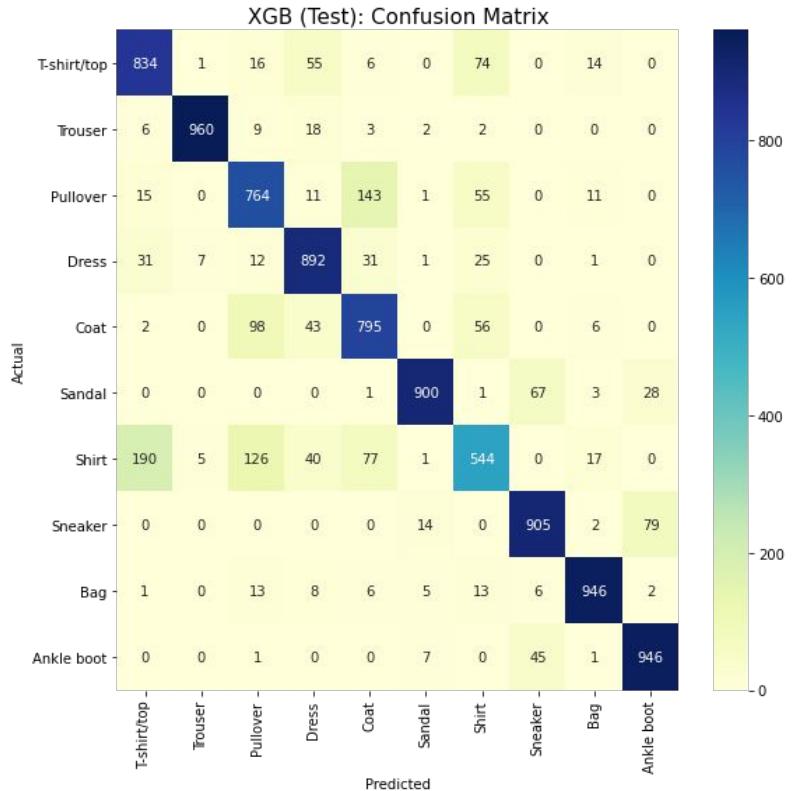
SVM - SVC



SVM RBF (Test)

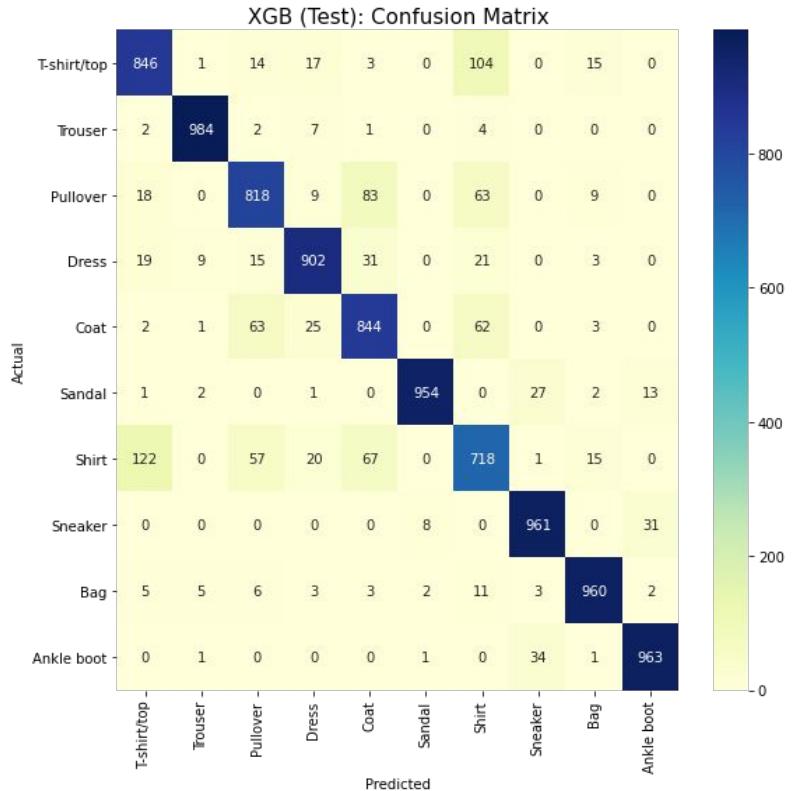
	precision	recall	f1-score	support
T-shirt/top	0.80	0.84	0.82	1000
Trouser	0.99	0.97	0.98	1000
Pullover	0.80	0.78	0.79	1000
Dress	0.87	0.91	0.89	1000
Coat	0.80	0.84	0.82	1000
Sandal	0.95	0.92	0.93	1000
Shirt	0.70	0.62	0.66	1000
Sneaker	0.91	0.93	0.92	1000
Bag	0.96	0.96	0.96	1000
Ankle boot	0.94	0.94	0.94	1000
accuracy			0.87	10000
macro avg	0.87	0.87	0.87	10000
weighted avg	0.87	0.87	0.87	10000

XGBoost Classifier



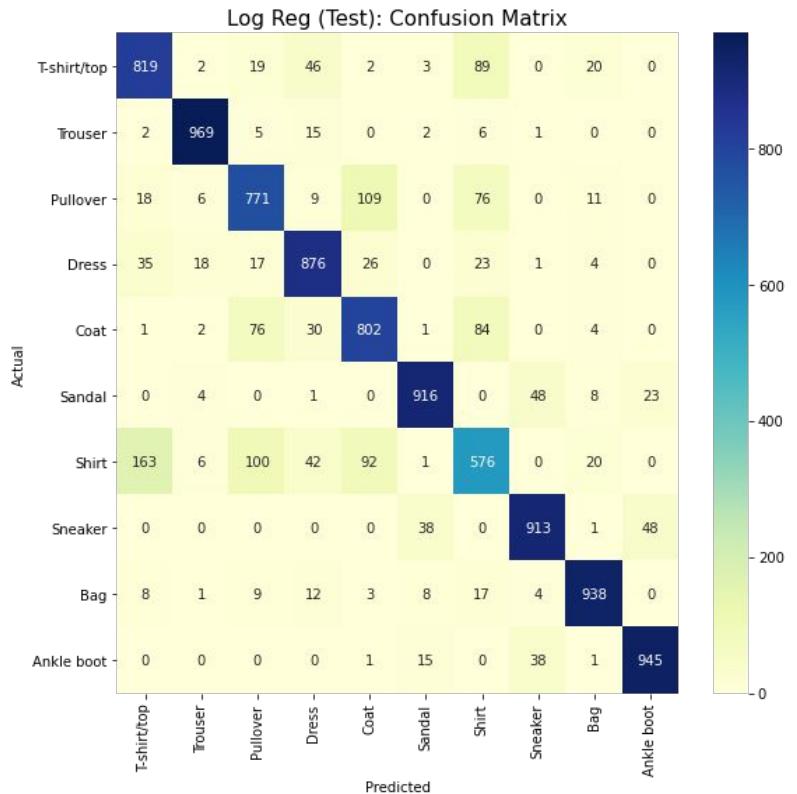
XGB (Test)	precision	recall	f1-score	support
T-shirt/top	0.77	0.83	0.80	1000
Trouser	0.99	0.96	0.97	1000
Pullover	0.74	0.76	0.75	1000
Dress	0.84	0.89	0.86	1000
Coat	0.75	0.80	0.77	1000
Sandal	0.97	0.90	0.93	1000
Shirt	0.71	0.54	0.61	1000
Sneaker	0.88	0.91	0.89	1000
Bag	0.95	0.95	0.95	1000
Ankle boot	0.90	0.95	0.92	1000
accuracy			0.85	10000
macro avg	0.85	0.85	0.85	10000
weighted avg	0.85	0.85	0.85	10000

Gradient Boost Classifier



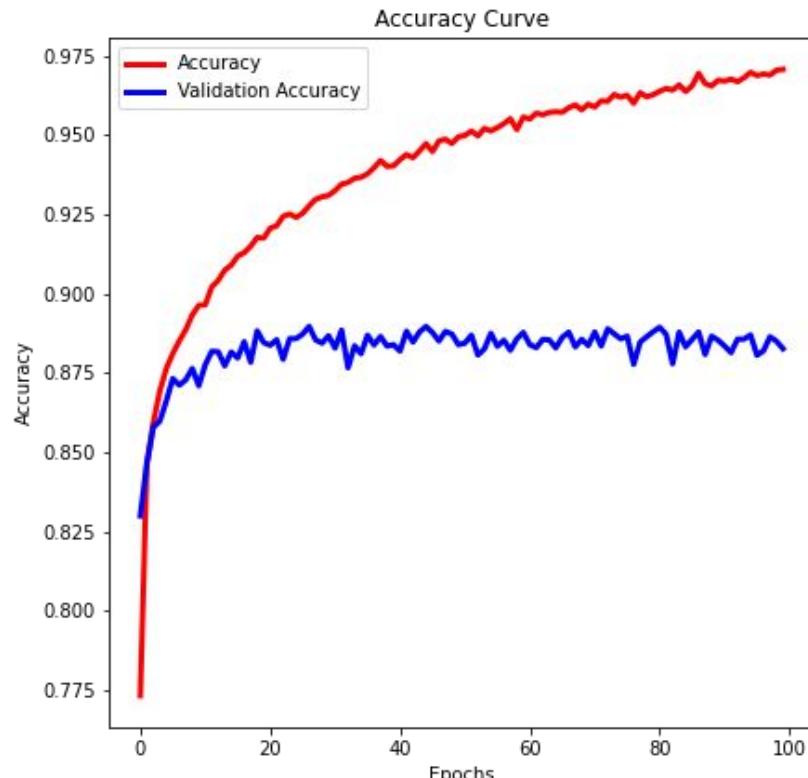
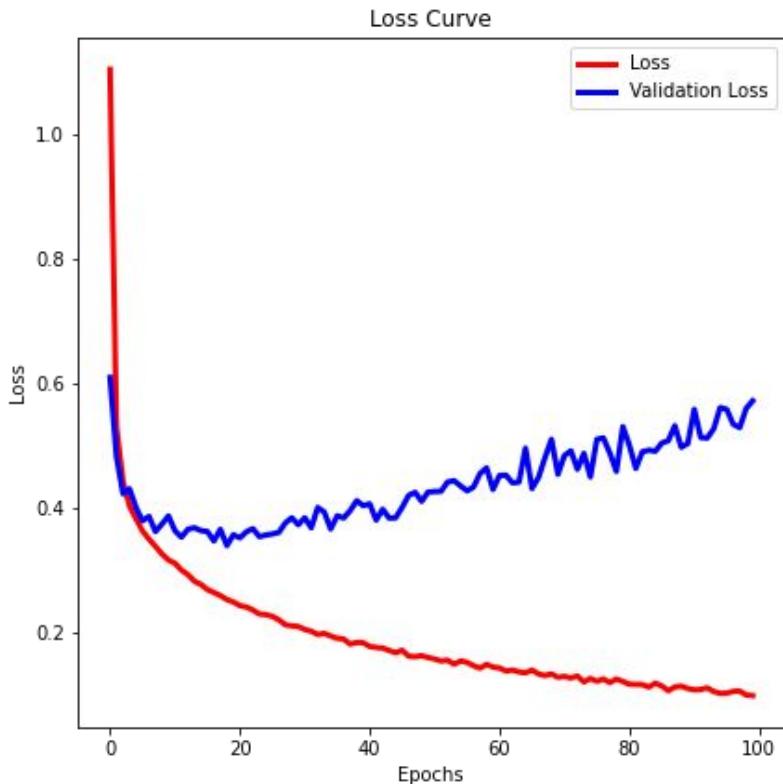
GBC (Test)	precision	recall	f1-score	support
T-shirt/top	0.83	0.85	0.84	1000
Trouser	0.98	0.98	0.98	1000
Pullover	0.84	0.82	0.83	1000
Dress	0.92	0.90	0.91	1000
Coat	0.82	0.84	0.83	1000
Sandal	0.99	0.95	0.97	1000
Shirt	0.73	0.72	0.72	1000
Sneaker	0.94	0.96	0.95	1000
Bag	0.95	0.96	0.96	1000
Ankle boot	0.95	0.96	0.96	1000
accuracy			0.90	10000
macro avg	0.90	0.89	0.89	10000
weighted avg	0.90	0.90	0.89	10000

Logistic Regression

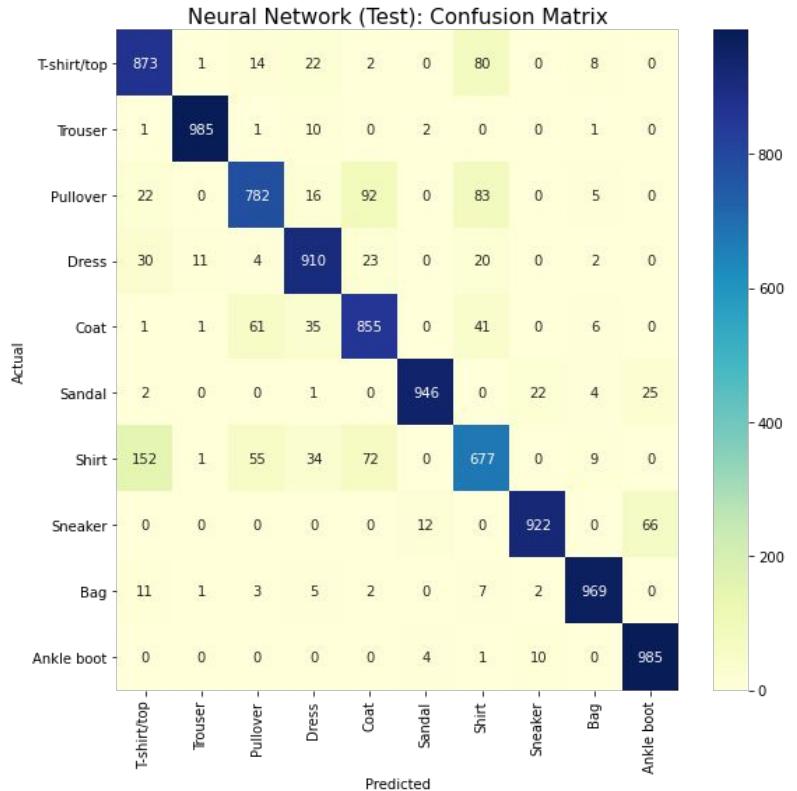


	precision	recall	f1-score	support
T-shirt/top	0.78	0.82	0.80	1000
Trouser	0.96	0.97	0.97	1000
Pullover	0.77	0.77	0.77	1000
Dress	0.85	0.88	0.86	1000
Coat	0.77	0.80	0.79	1000
Sandal	0.93	0.92	0.92	1000
Shirt	0.66	0.58	0.62	1000
Sneaker	0.91	0.91	0.91	1000
Bag	0.93	0.94	0.93	1000
Ankle boot	0.93	0.94	0.94	1000
accuracy			0.85	10000
macro avg	0.85	0.85	0.85	10000
weighted avg	0.85	0.85	0.85	10000

Simple Neural Network



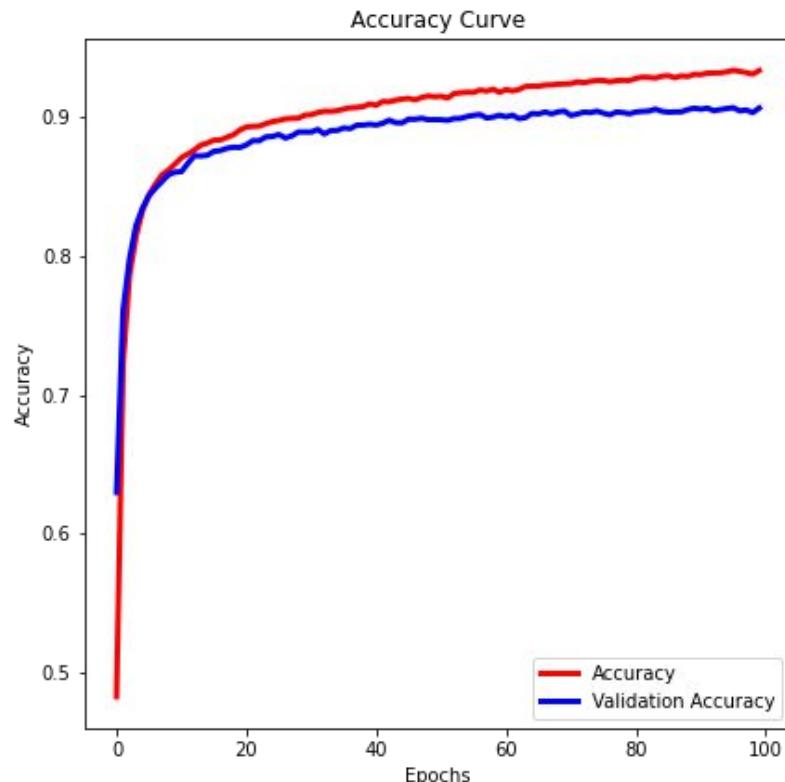
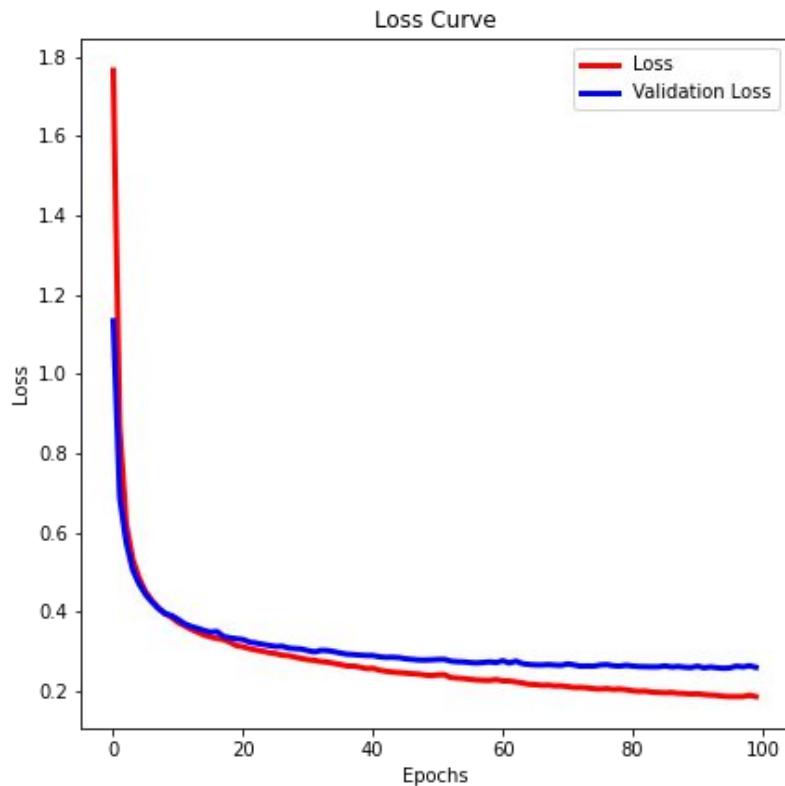
Simple Neural Network



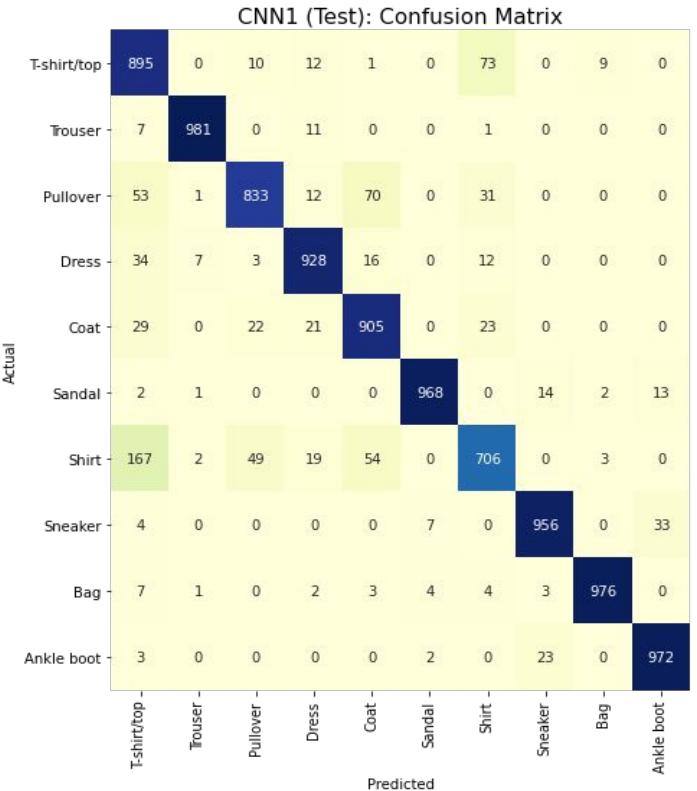
NN (Test)

	precision	recall	f1-score	support
T-shirt/top	0.80	0.87	0.83	1000
Trouser	0.98	0.98	0.98	1000
Pullover	0.85	0.78	0.81	1000
Dress	0.88	0.91	0.90	1000
Coat	0.82	0.85	0.84	1000
Sandal	0.98	0.95	0.96	1000
Shirt	0.74	0.68	0.71	1000
Sneaker	0.96	0.92	0.94	1000
Bag	0.97	0.97	0.97	1000
Ankle boot	0.92	0.98	0.95	1000
accuracy			0.89	10000
macro avg	0.89	0.89	0.89	10000
weighted avg	0.89	0.89	0.89	10000

Neural Network with 1 CNN Layer

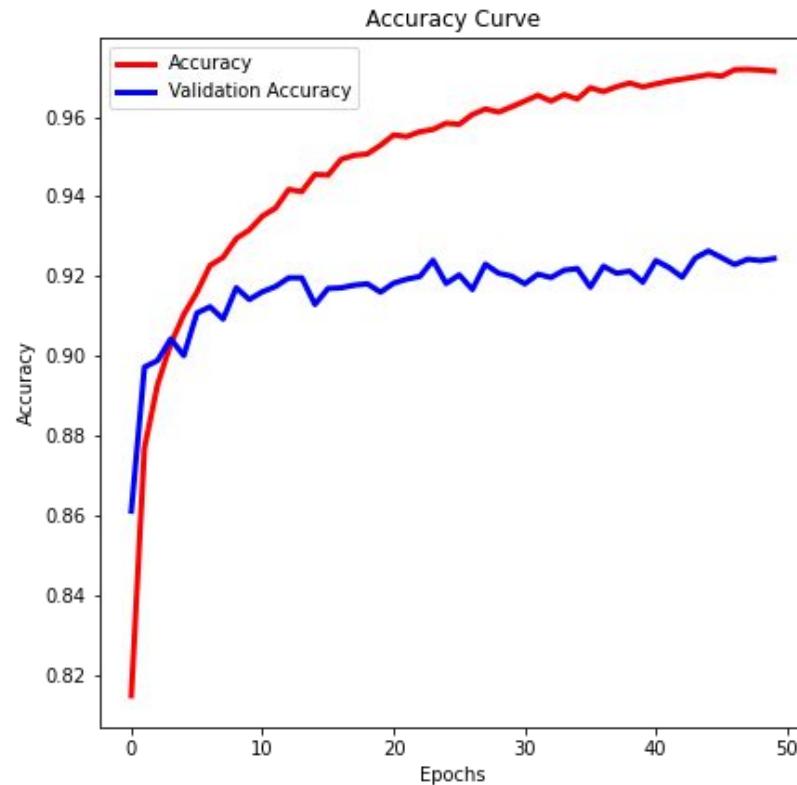
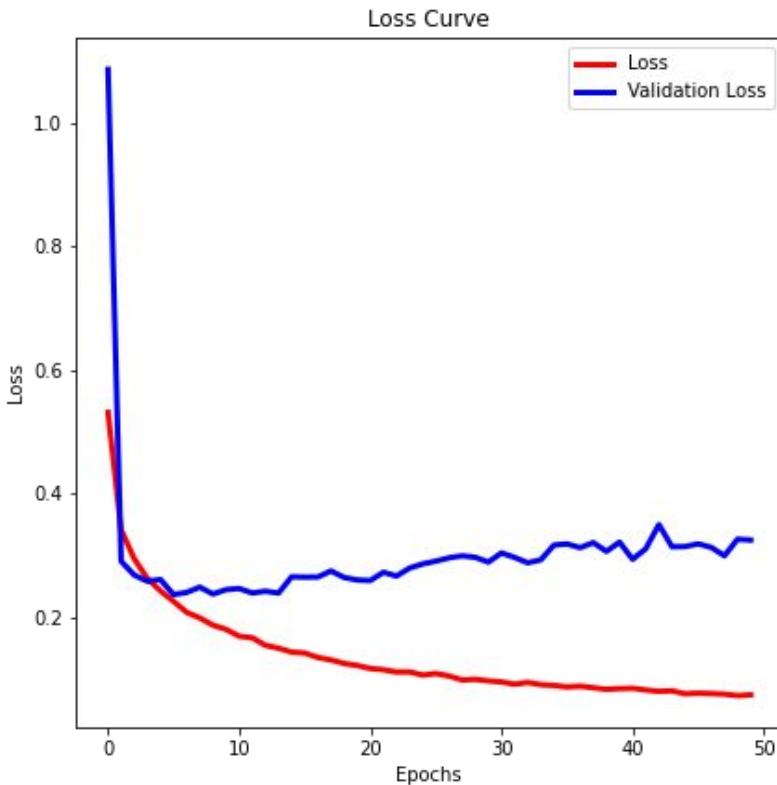


Neural Network with 1 CNN Layer

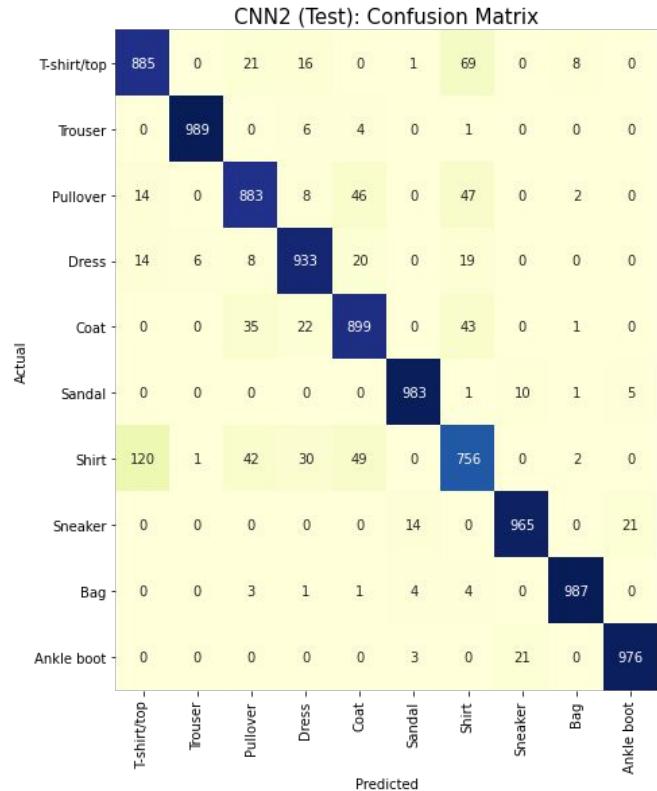


	CNN 1 (Test)				
	precision	recall	f1-score	support	
0	0.75	0.90	0.81	1000	
1	0.99	0.98	0.98	1000	
2	0.91	0.83	0.87	1000	
3	0.92	0.93	0.93	1000	
4	0.86	0.91	0.88	1000	
5	0.99	0.97	0.98	1000	
6	0.83	0.71	0.76	1000	
7	0.96	0.96	0.96	1000	
8	0.99	0.98	0.98	1000	
9	0.95	0.97	0.96	1000	
accuracy				0.91	10000
macro avg	0.91	0.91	0.91	10000	
weighted avg	0.91	0.91	0.91	10000	

Neural Network with 2 CNN Layers



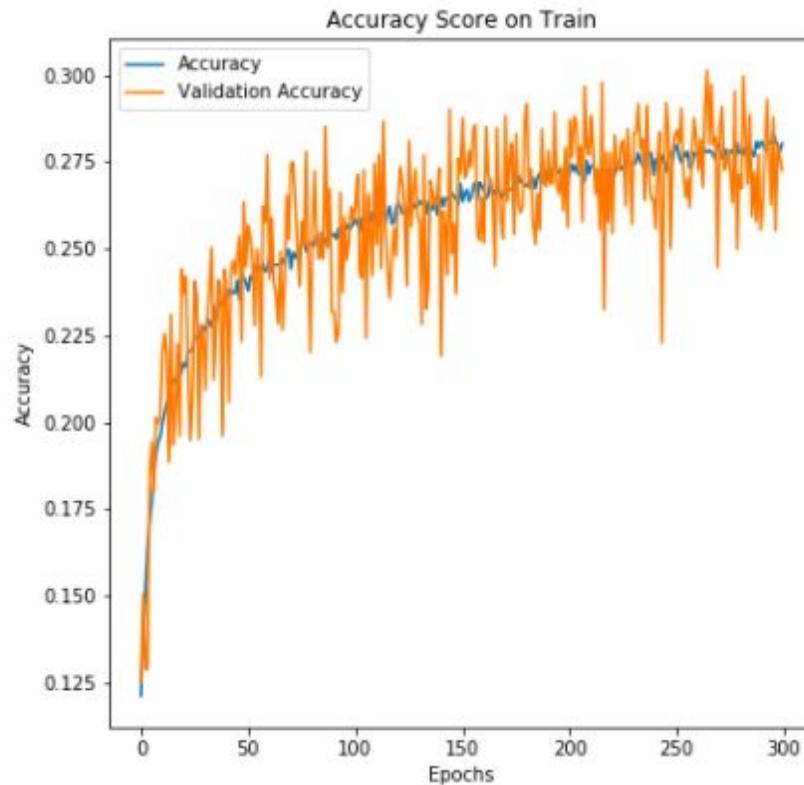
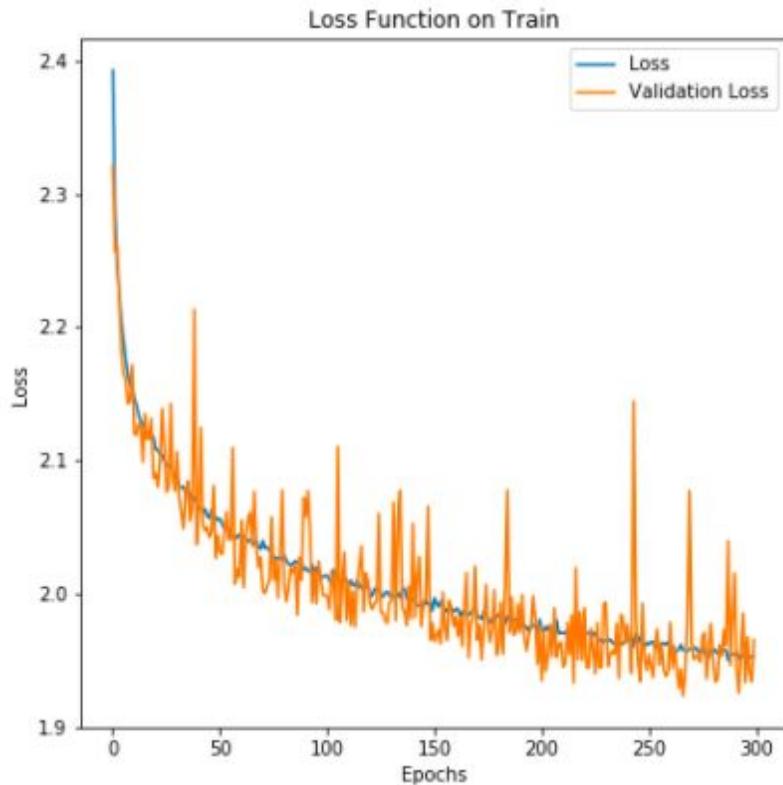
Neural Network with 2 CNN Layers



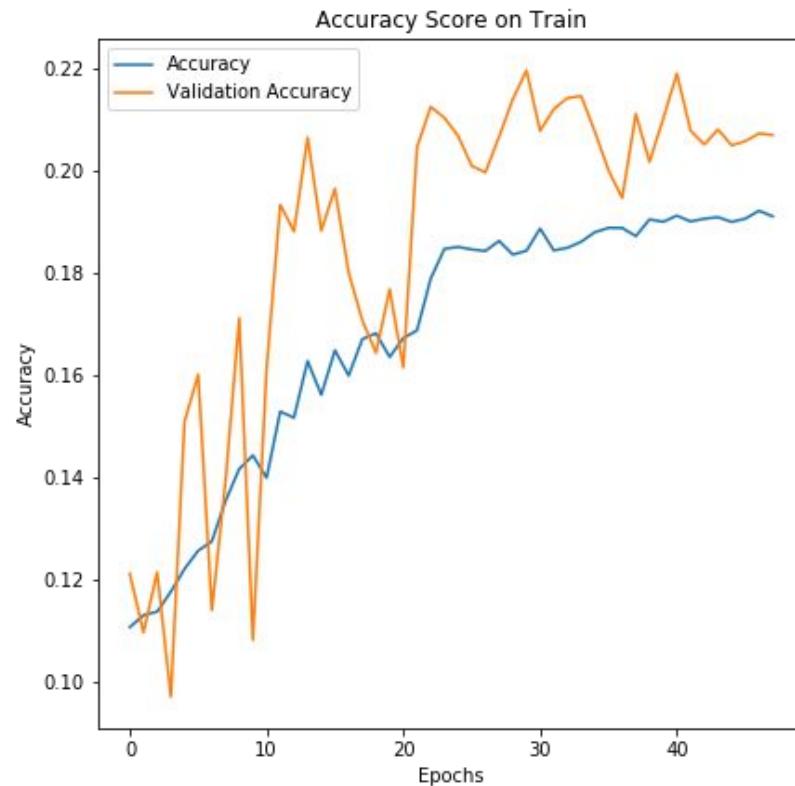
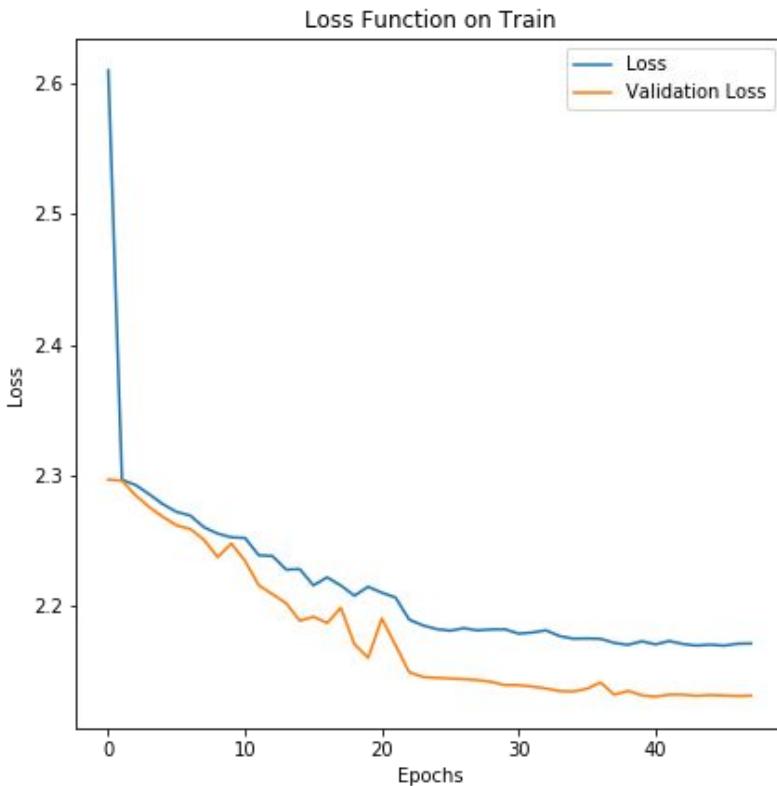
CNN 2 (Test)

	precision	recall	f1-score	support
0	0.86	0.89	0.87	1000
1	0.99	0.99	0.99	1000
2	0.89	0.88	0.89	1000
3	0.92	0.93	0.93	1000
4	0.88	0.90	0.89	1000
5	0.98	0.98	0.98	1000
6	0.80	0.76	0.78	1000
7	0.97	0.96	0.97	1000
8	0.99	0.99	0.99	1000
9	0.97	0.98	0.98	1000
accuracy			0.93	10000
macro avg	0.93	0.93	0.93	10000
weighted avg	0.93	0.93	0.93	10000

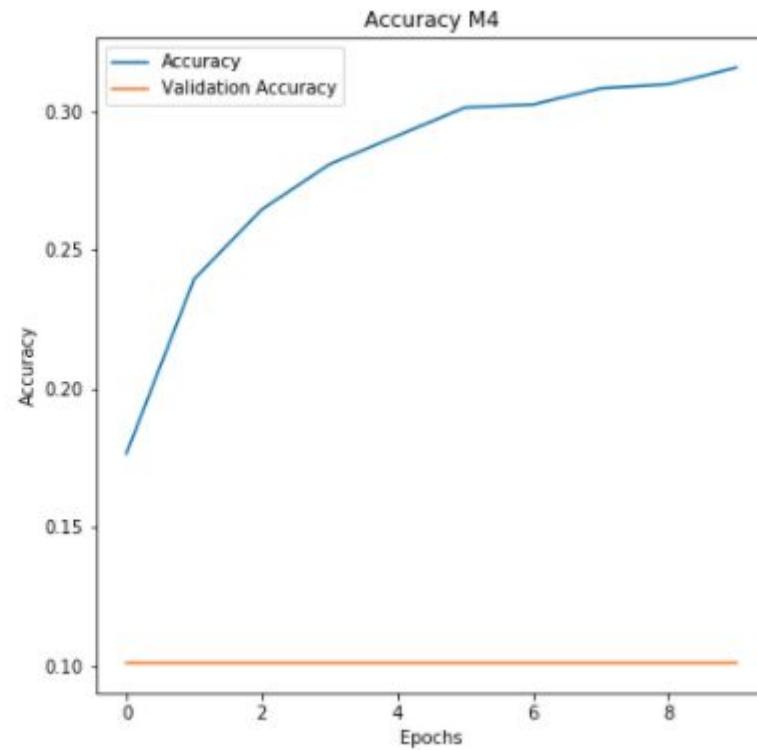
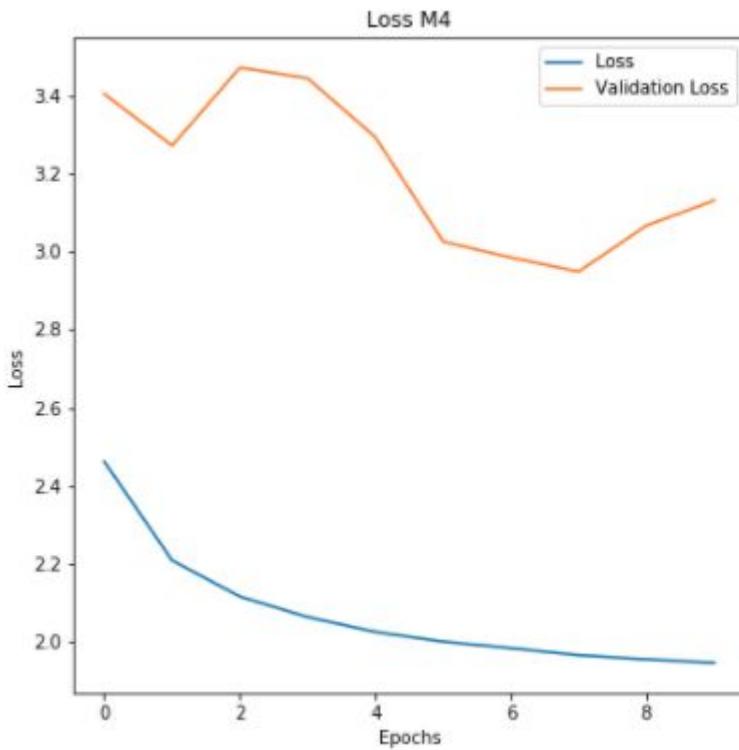
VGG 16



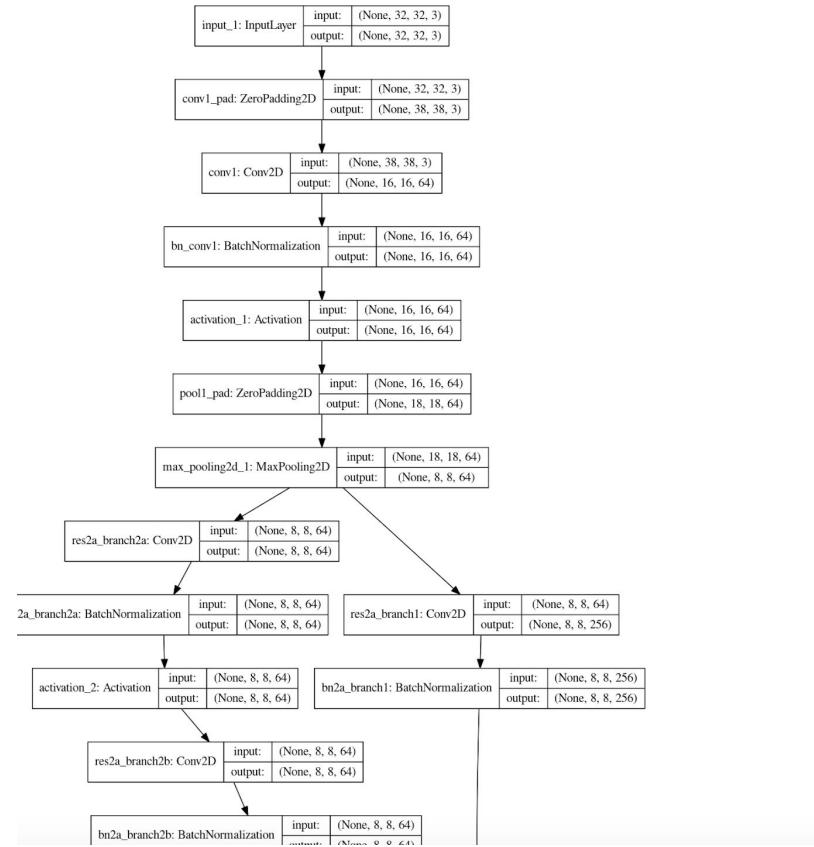
VGG 19



ResNet50

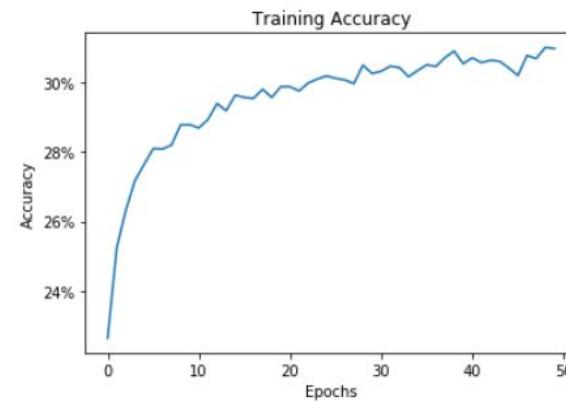
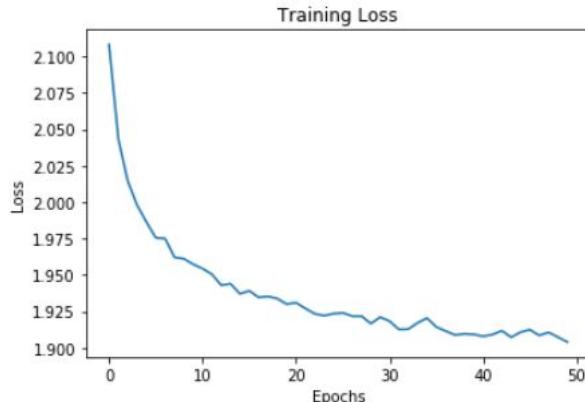


ResNet50 - Beginning Layers



InceptionV3

```
Epoch 46/50  
60000/60000 [=====] - 410s 7ms/step - loss: 1.9125 - accuracy: 0.3019  
Epoch 47/50  
60000/60000 [=====] - 410s 7ms/step - loss: 1.9086 - accuracy: 0.3076  
Epoch 48/50  
60000/60000 [=====] - 410s 7ms/step - loss: 1.9107 - accuracy: 0.3067  
Epoch 49/50  
60000/60000 [=====] - 410s 7ms/step - loss: 1.9074 - accuracy: 0.3099  
Epoch 50/50  
60000/60000 [=====] - 410s 7ms/step - loss: 1.9041 - accuracy: 0.3096
```



InceptionResNetV2

```
Epoch 46/50  
4800/4800 [=====] - 111s 23ms/step - loss: 1.6861 - accuracy: 0.4137  
Epoch 47/50  
4800/4800 [=====] - 112s 23ms/step - loss: 1.6552 - accuracy: 0.4125  
Epoch 48/50  
4800/4800 [=====] - 111s 23ms/step - loss: 1.6936 - accuracy: 0.4112  
Epoch 49/50  
4800/4800 [=====] - 111s 23ms/step - loss: 1.6862 - accuracy: 0.4042  
Epoch 50/50  
4800/4800 [=====] - 112s 23ms/step - loss: 1.7190 - accuracy: 0.3977
```

